Spring 2016

Plan B

Sriyakar Reddy Baddam
Governors State University

Kranthi Kumar Reddy Kallem
Governors State University

Durga Prasad Kosuri
Governors State University

Table of Contents

1 Project Description

1.1 Project Abstract

1.1.1 Motivation:

For any person with his/her hectic schedule in today's world, planning and executing a gathering with friends or family is becoming complicated. At times we cancel a plan due to the befuddlement caused while coordinating people. We are planning to resolve this problem with the magic of technology that gives solution to all the problems. The project "Plan B", an android application uses simple and effective design to help users plan and execute a gathering. This application needs a user to register and login with the primary email id which he/she uses such that the gathering planned through Plan B can check with the already existing events in calendar.

1.1.2 Uniqueness:

The application "Plan B" is unique from the other applications in the market which helps in organizing gatherings, because it allows you to create groups and share the planned event directly with the group. It also checks the available schedule of the people from the registered email id's calendar and suggest a new time if there is any time conflict. Based on the user's rating about the previous visits the application will give suggestions in future.

1.1.3 System Features:

OS	: Windows/iOS
IDE	: Android Studio
MVC	: Visual Studio
Languages	: Java, J2EE
Back end	: Apache Spark
Database	: MongoDB

1.1.4 Objectives:

The main objective of this project is to build an android application which can help people plan an event or hangout based on all the available time slots of each and every one participating in the event and come up with an optimized time slot. In this way the communication between each and every one participating will be easier and saves a lot of time and cuts down unnecessary discussion. Based on the user's likings the application can come up with suggestions in future, such as movie recommendation, a new restaurant opened.

1.1.5 Related work:

- Tossup application provides a user to create gathering for a dinner or drink.
- Yelp application provides all the available restaurants and shopping centers based on a user's current location.

1.2 Competitive Information

As a part of competitive information we don't think there is application like this .The features that we developed, designed and technology we used makes us to be competitive for other project. I think we have that potential to market our project. We have a great confidence that we can withstand in the market and we can market our project.

1.3 Relationship to Other Applications/Projects

We have some relationships and some unique objects that make us different from other application like our API, services and some frameworks we used for developing our project. As a part of design we took base from some online applications and we developed with our thoughts.

1.4 Assumptions and Dependencies

- The Mobile application is dependent on web application because mobile application can't run without the data from the web application and vice versa
- The default setting on the mobile application will reflect on the google maps
  Settings such as frequently used zip codes and the radius can be made default, thereby providing easy way for the service provider.
  Each service provider
- Each service provider having a Google android phone
  Each service Provider using the application shall process a smart phone and with android operating system on top of it

1.5 Future Enhancements

As a part of our future plan we want to extend our project and we want to release in to the market .As a part of some iterations we have developed to an basic version .To work more on this project we need some time for testing and to fix bugs and we have also provided bug fixing updates and version updates.

1.6 Definitions and Acronyms

- AngularJS----It aims to simplify both the development and the testing
- DFD -----Data Flow Diagram
- CFD -----Context Flow Diagram
- ER ------Entity Relationship
- IDE ------ Integrated Development Environment
- SRS ------ Software Requirement Specification

2 Technical Description

2.1 Ionic Framework: We used Ionic framework for the mobile application development as it is a open-source sdk. Ionic helps to develop the mobile applications using web technologies such as HTML, CSS by providing tools and services. Ionic provide all
Functionalities that are provided by native mobile application development. User can customize their app to android or ios using Ionic.

2.2 Design Patterns: Design patterns are used for solving generally occurring problems in the application development. we can reuse these designs using patterns. We have used a singleton design pattern for a simplified base URL to be used commonly for accessing mongoLab by the servlets and controllers.

2.3 ngCordova: ngCordova is a powerful tool made of angularjs facilitates for the faster and efficient development of the mobile applications. It provides angularjs wrappers to the cordova plugins. We have used a few plugins such as n gCordovaCalendar,
ngCordova-contacts, ngCordova- google plus login, ngCordova-facebook into our
project.

2.4 MongoLab: We are using MongoDB in MongoLab as a database to store our user credentials,messages, events and friend lists. we connect to the mongolab using api and url. we create a mongoclient using the url and api and access the data using CURD operations.Give a brief overview of project design and describe the technical make-up of the project. Remember, the main reason for writing this document is to provide sufficient details for your reader and/or the developers who will implement this project. In this case, it will be your own team

2.1 Project/Application Architecture

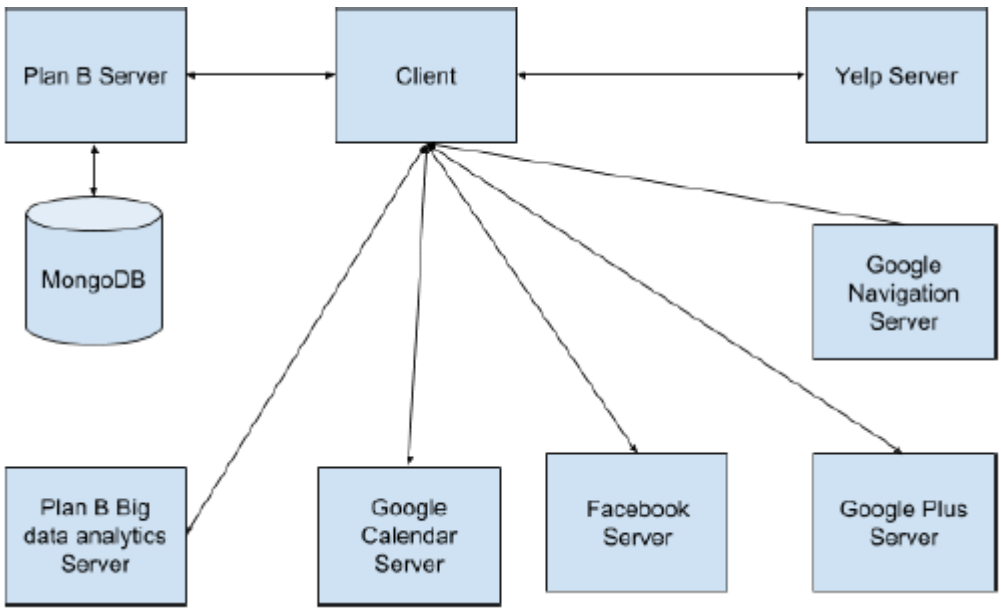

Figure 1.1

2.2 Project/Application Information flows

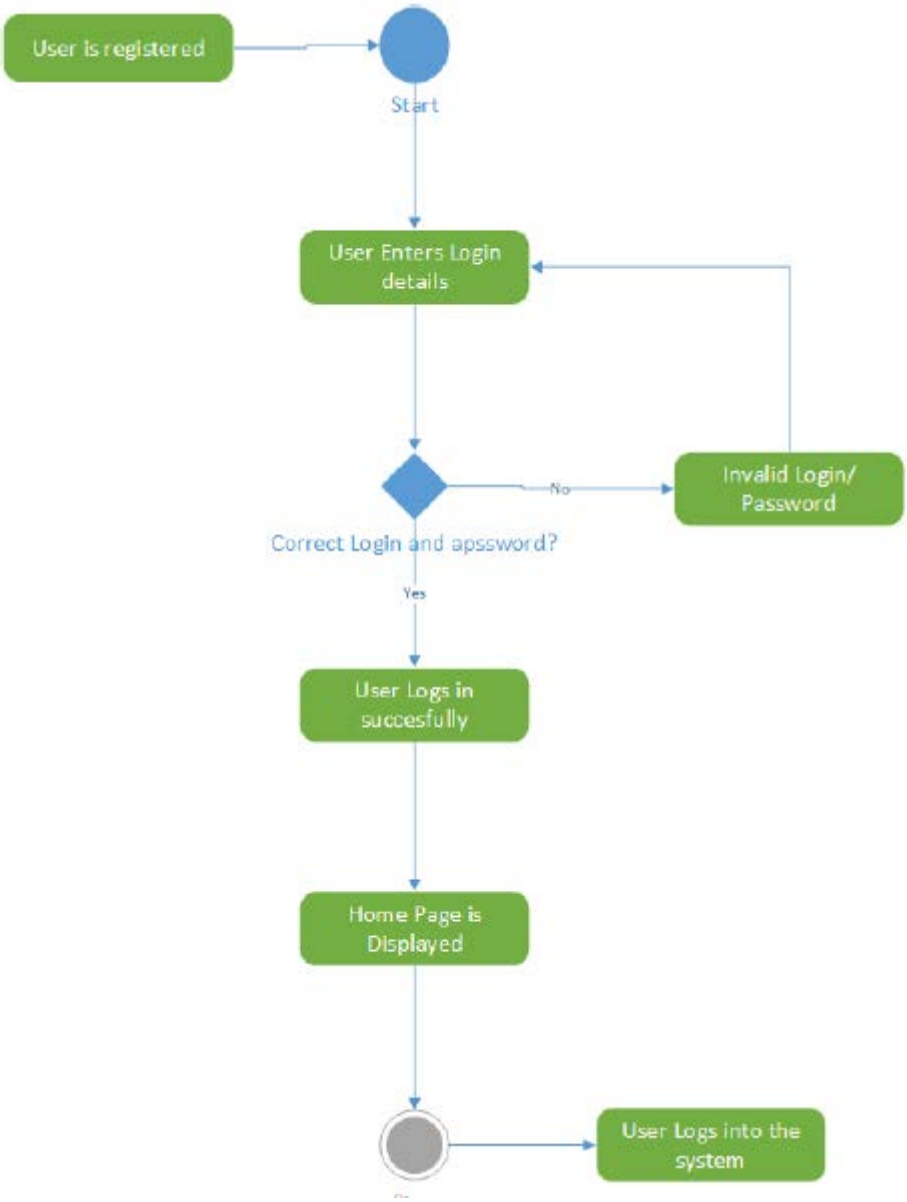

Figure 1.2

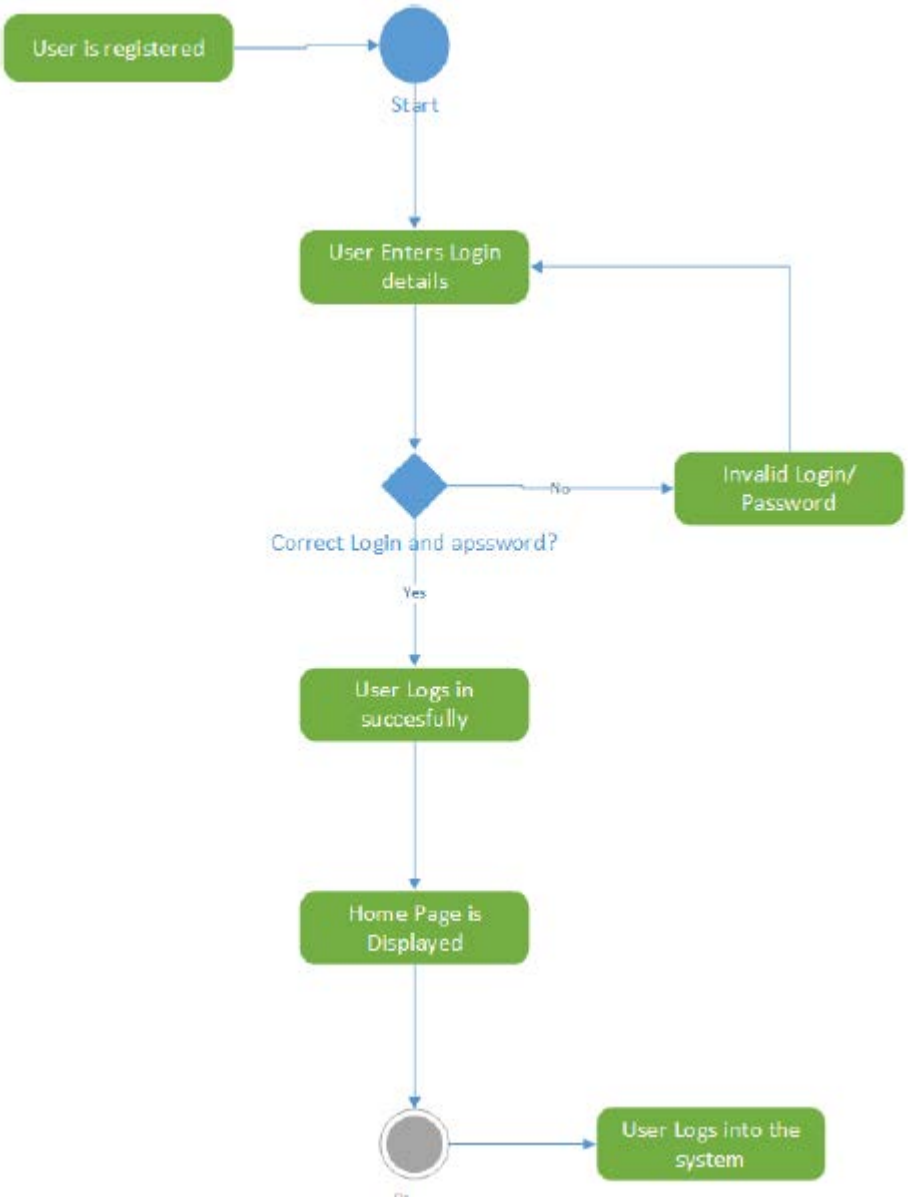
User is registered
Start
User Enters Login details
No
Invalid Login/ Password
Correct Login and apssword?
Yes
User Logs in succesfully
Home Page is Displayed
User Logs into the system

Figure 1.3

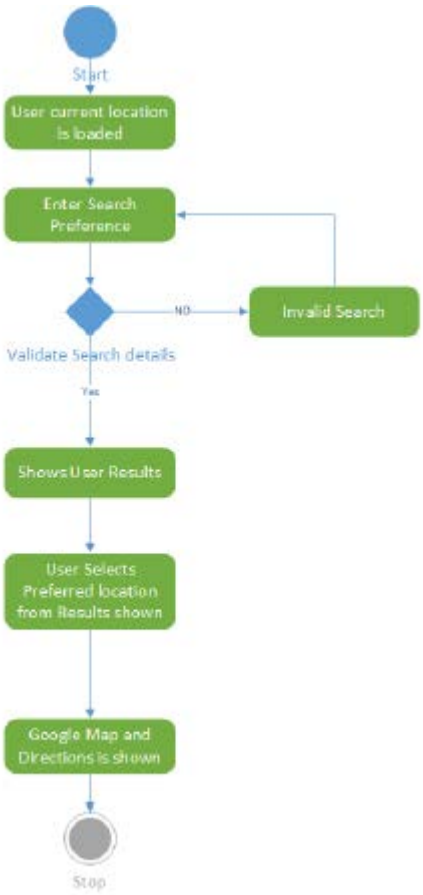

Figure 1.4

2.3 Interactions with other Projects (if Any)

2.4 Interactions with other Applications

We used the help of below services to implement in our project

● Yelp API : Searching relevant local food chains , events and getting rating for the same

● Google Direction API : Directions to chosen local food chain.

Identified Primary Services :

● Facebook API: For Login Facilitation, Updating Events Calendar and Check‐ins. Retrieving Friends List to add onto the contacts in our application

● Google Plus API : For Login Facilitation, Updating Events Calendar and Google‐Now Notification

● Mongolab API : For making all data more accessible rather than local MongoDB

● Google Calendars : For adding and validating for possible upcoming events

● Eventful API : For upcoming events and happenings of the city

● Twitter API : Providing the user an insight about the happening trends about the

2.5 EventsCapabilities

We have developed the following features in order to achieve the proposed project.

● Cordova contacts plugin

● Google Maps API

● checking the events list

● messaging between users

● Creating friends list manually

● Message service to send invitation to friends

● Cordova Calendar plugin

2.6 Risk Assessment and Management

Having a secure password having alphanumeric numbers in the password will be more effective.

3 Project Requirements

3.1 Identification of Requirements

Register a new user

● Login with the UserId and Password created

● Stored user details in MongoDB

● Search restaurants using Yelp API

● Get directions to the event from current location using Google directions

○ Method used: google.maps .DirectionsService())

A profile page for the user

● Settings page for the user

● Cordova contacts plugin to add the user to mongolab

● Google maps API in order to locate the location of the event.

● User must be able to check his event list by clicking a button and must be able to delete the event

● User can update his contacts list by editing and deleting his contacts using the username of the contact.

● Messaging between users. users can create the messages using the respective usernames and the messages will be stored in database and the corresponding user can view his messages.

● Creating friends list manually by providing his contact details.

● Our application uses the messaging service(SMS) to send the invitation to the user.

● Cordova calendar plugin is used in order to check user's schedule

Register a new user

- Login with UserId and Password or facebook or google plus for an existing user
- Integrated Cordova Facebook Login plugin
- Integrated Cordova Google Plus Login plugin
- Integrated Cordova Contact plugin to access phone contacts
- Stored user details and event created details in MongoDB(MongoLab

3.2 Operations, Administration, Maintenance and Provisioning (OAM&P)

MongoDB Authorization

MongoDB allows administrators to define the specific permissions an application or user has, and what data they can see when querying the database.

3.2.1 MongoDB Auditing

The MongoDB Enterprise Advanced auditing framework logs all access and actions executed against the database. The auditing framework captures administrative actions (DDL) such as schema operations as well as authentication and authorization activities, along with read and write (DML) operations to the database. Administrators can construct and filter audit trails for any operation against MongoDB, whether DML, DCL or DDL without having to rely on third party tools. For example, it is possible to log and audit the identities of users who retrieved specific documents, and any changes made to the database during their session

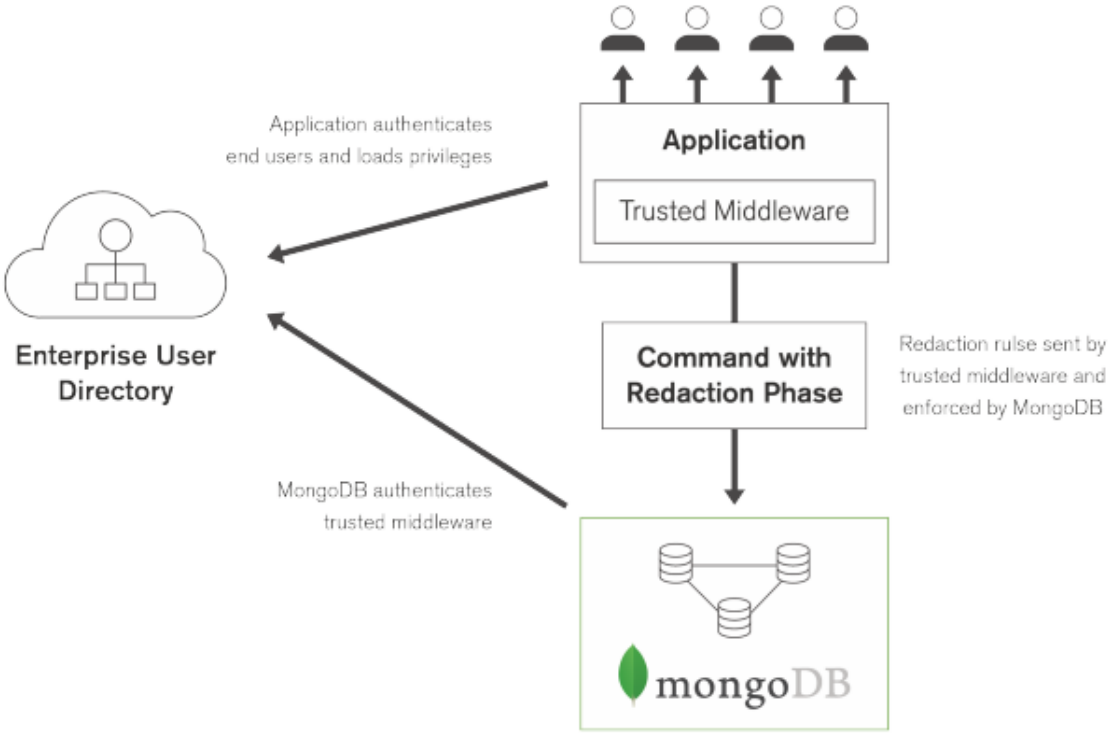

Figure 1.5

3.2.2 MongoDB Encryption

Administrators can encrypt MongoDB data in motion over the network and at rest in permanent storage.

3.2.3 Network Encryption

Support for SSL/TLS allows clients to connect to MongoDB over an encrypted channel. Clients are defined as any entity capable of connecting to the MongoDB server, including: • Users and administrators • Applications • MongoDB tools (e.g., mongodump, mongorestore, mongotop) • Nodes that make up a MongoDB cluster, such as replica set members, query routers and config servers.

3.3 Security and Fraud Prevention

Secure environments use the following strategies to control access, with more detail available in the Network Exposure and Security section of the documentation.

3.3.1 Network Filter.

By using filters such as firewalls and router ACL rules, connections to MongoDB from unknown systems can be blocked. Firewalls should limit both incoming and outgoing traffic to/from a specific port to trusted and untrusted systems. For best results and to minimize overall exposure, ensure that only traffic from trusted sources can reach mongod and mongos instances and that the mongod and mongos instances can only connect to trusted outputs. In addition, unneeded system services should be deactivated.

3.3.2 Binding IP Addresses. The bind_ip setting for mongod and mongos instances limits the network interfaces on which MongoDB programs will listen for incoming connections.

3.3.3 Running in VPNs.

Limit MongoDB programs to non-public local networks and virtual private networks. Virtual Private Networks (VPNs) make it possible to link two networks over an encrypted and limited-access trusted network. Typically MongoDB users configure SSL rather than IPSEC protocols for performance advantages.

3.3.4 Dedicated OS User Account.

A user account dedicated to MongoDB should be created and used to run MongoDB executables. MongoDB should not run as the “root” user.

3.3.5 File System Permissions.

The servers running MongoDB should employ filesystem permissions that prevent users from accessing the data files created by MongoDB. MongoDB configuration files and the cluster keyfile should be protected to disallow access by unauthorized users.

3.3.6 Query Injection.

As a client program assembles a query in MongoDB, it builds a BSON object, not a string. Thus traditional SQL injection attacks should not pose a risk to the system for queries submitted as BSON objects.

3.4 Release and Transition Plan

We developed android application which is an open source we can install into our android phone and we can use this app. As part of marketing we developed this project for user’s usage we have provided many flexibility’s for the user and can also follow our user guide.

4 Project Design Description

Class Diagram

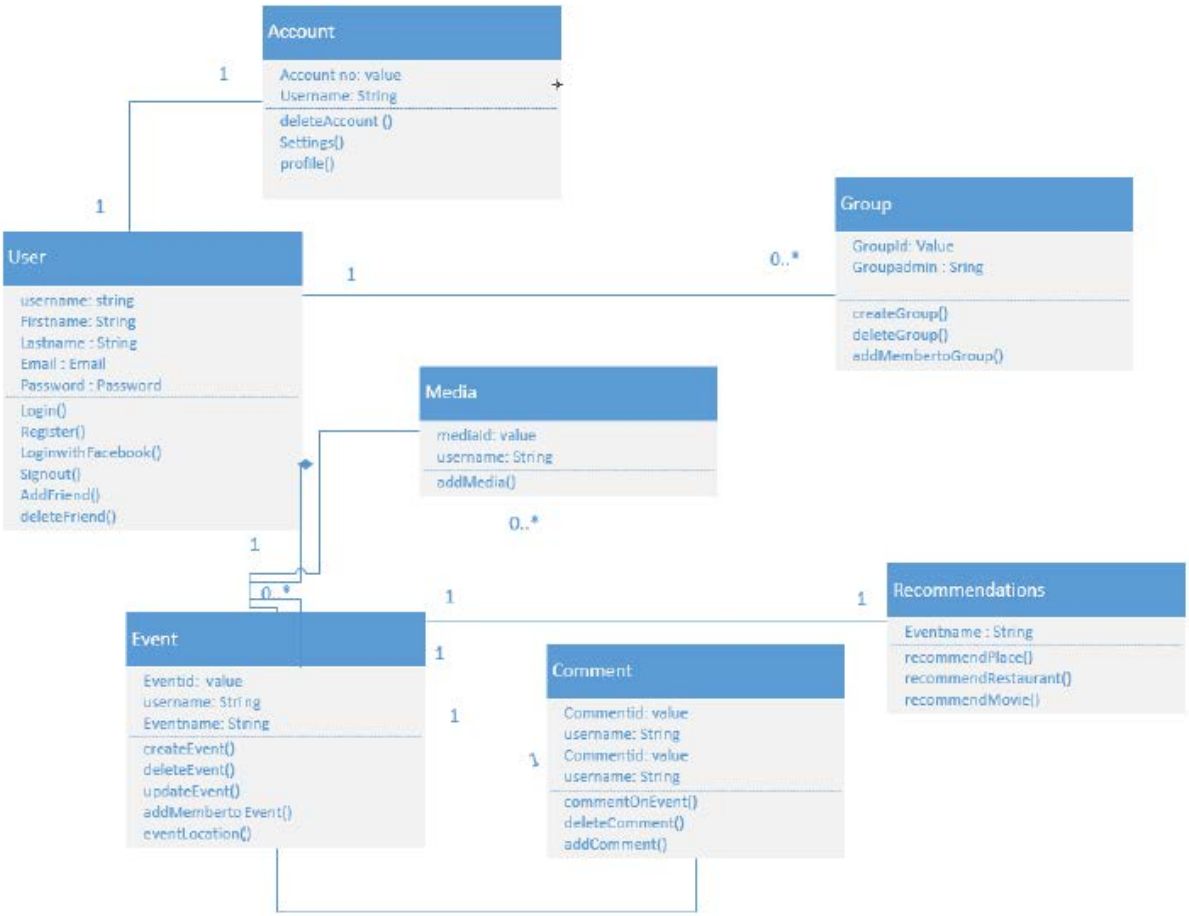

Figure 1.6

Activity Diagram

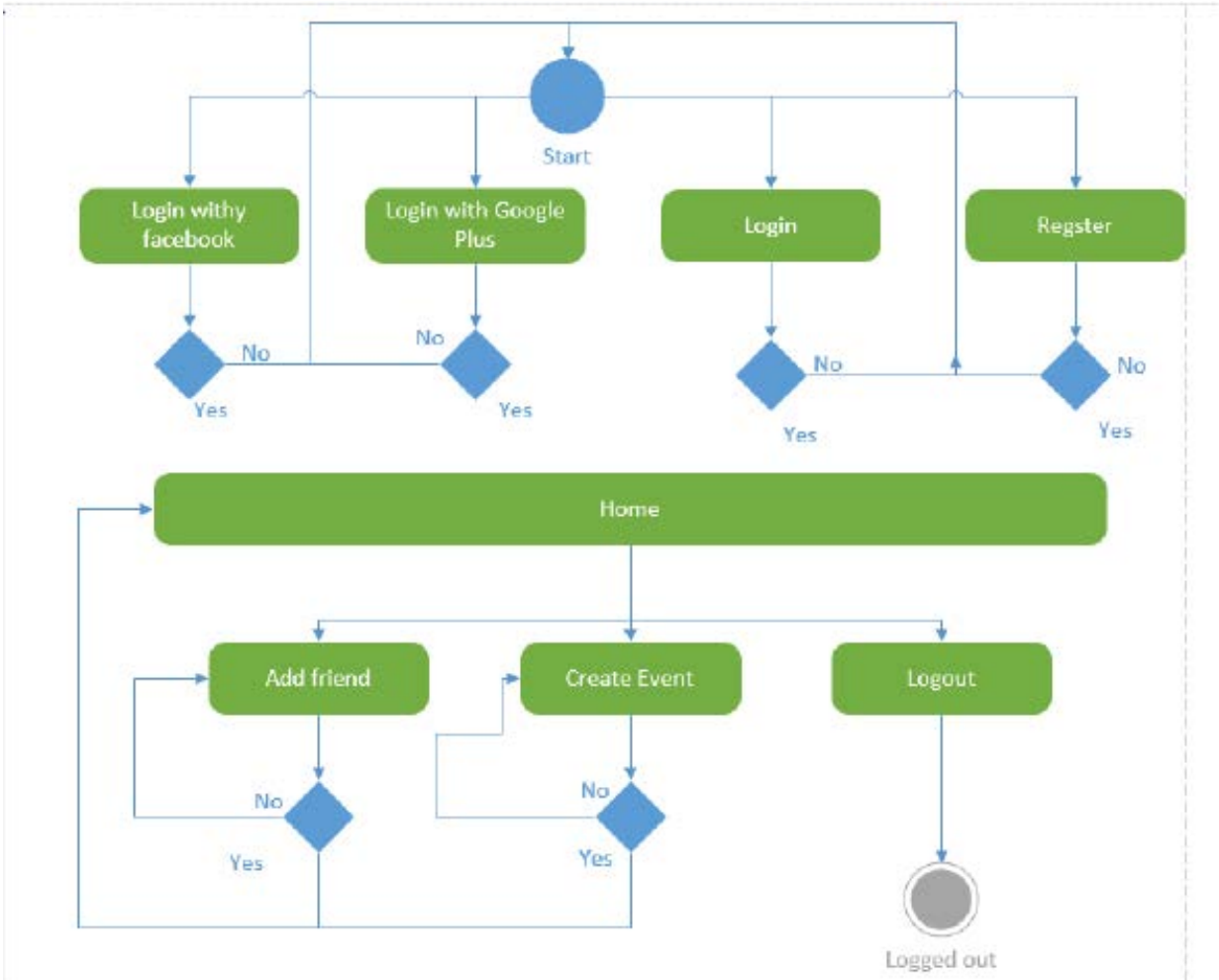

Figure 1.7

Sequence Diagram

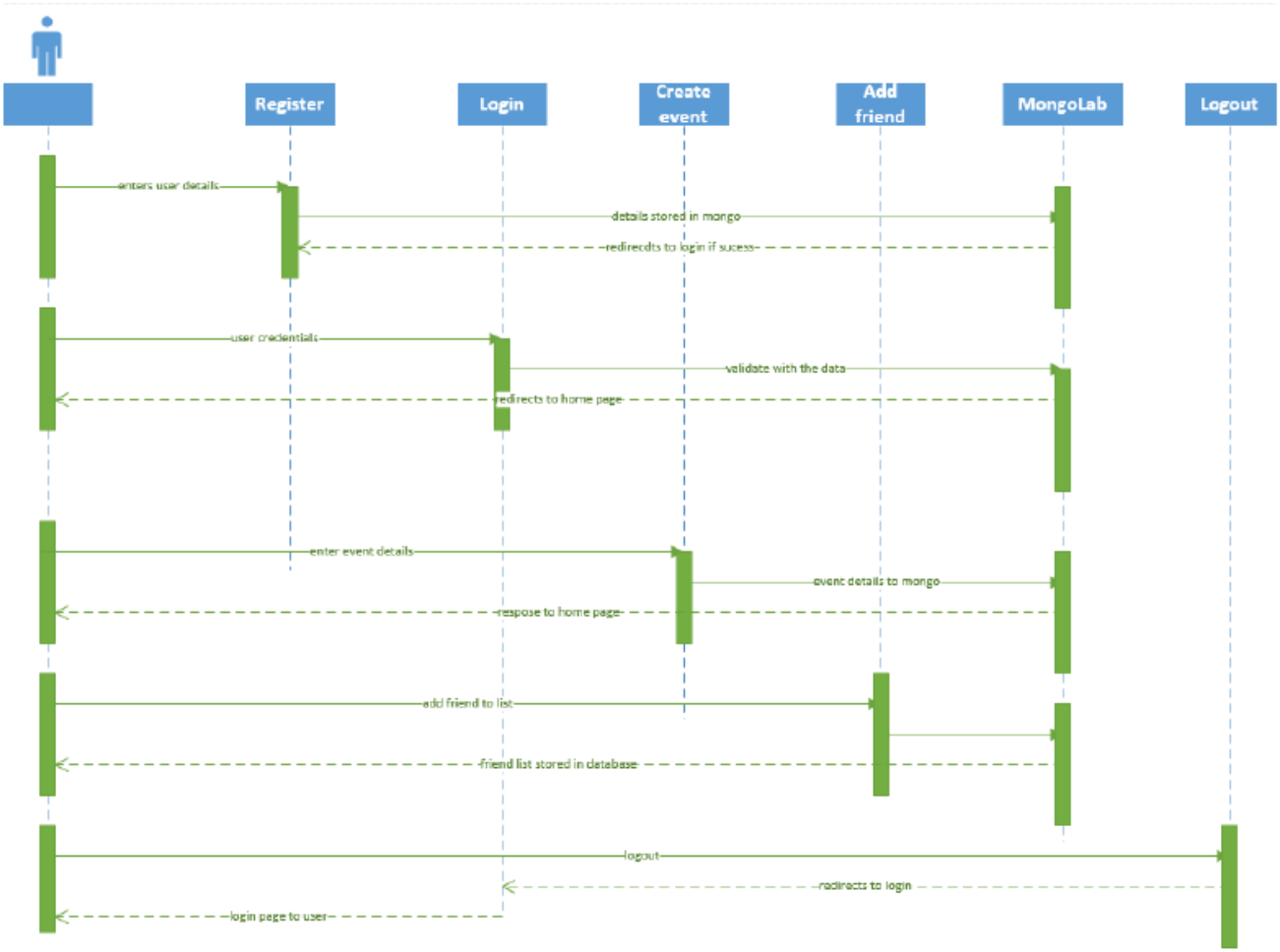

Figure 1.8

5 Project Internal/external Interface Impacts and Specification

We have provided flexible interface to the user which can be easily understandable. We have provide editable interface that can make changes in the database. User can make change like update his profile and delete. When he creates event he can make change he can also delete and update regarding his event

6 Project Design Units Impacts

The purpose of design phase is to plan a solution for problem specified by the requirements. System design aims is to identify the modules that should be in the system the specification of these modules and how they interact with each other to produce the desired result. The goal of

the design process is to produce a module or representation of a system which can be used later to build that system. The produced model is called design of the system.

The most important phase of the software of the system is designing the different modules. The accurate planning and proper interconnections with the modules will give a good output in the implementation part.

6.1 Functional Area/Design Unit A

We have developed a native android application for the proposed project andimplemented the above mentioned features. The architecture we have developed is Model- View- Controller (MVC), where MongoDB,Ionic framework,Servlets acts like M,V and C respectively . We have used Ionic framework for implementing the user side logic and develop the features, MongoDB to store and retrieve user and event details. Some REST services have been developed to insert,retrieve, update and delete data.

6.1.1 Functional Overview

Based on the requirement we have developed and designed based on the requirements and we used cordova and ionic framework for the development and to run this we used android tools.

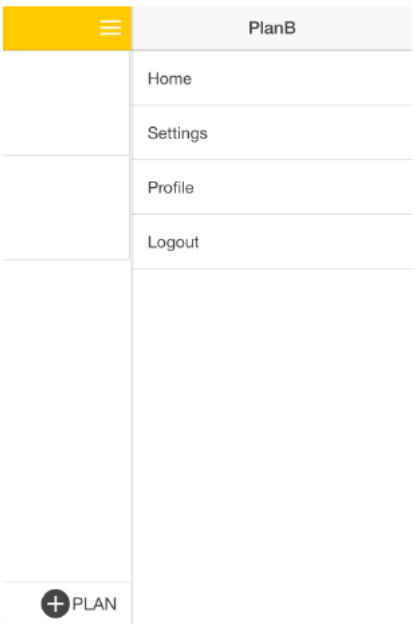

Figure 1.9

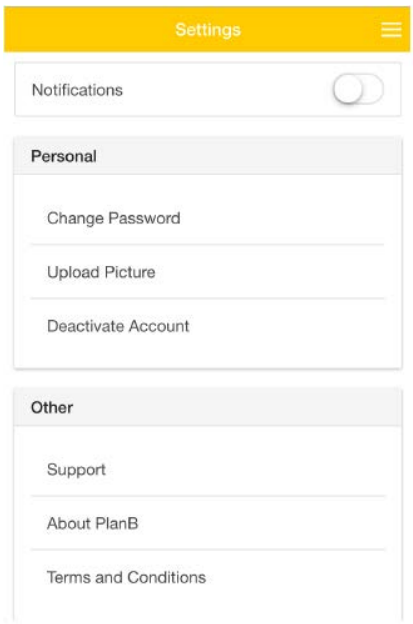

Figure 2.0

Figure 2.1

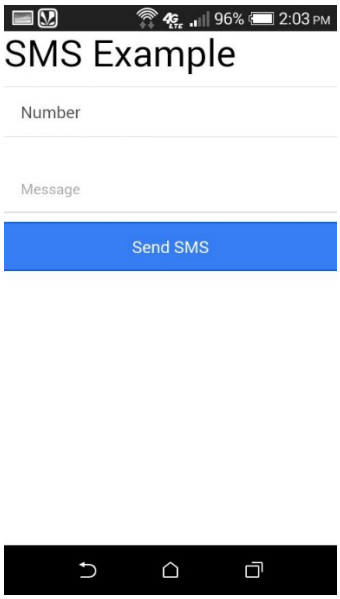

Figure 2.2

6.1.2 Impacts

We have used YSlow google chrome plugin to calculate the performance of our PlanB application. The YSlow application grades the application on a scale of A,B and C. YSlow has graded our application PlanB with “B”, “B” and “A” for different pages.

A/B Testing

We have developed two UI models and took inputs from more than 10 different individuals. We have collected their opinions and inputs about both models and finalized the most liked model..

6.1.3 Requirements

- . The Host logs into the application
- On the home screen, the user clicks on add event button.
- . The system transitions the user interface to the “Add Event” layout.
- . The user adds the title of the event, selects a list of friends he wants to invite and also selects date and time of the event and clicks on “Create” button.
- . System adds the events information to the database.
- . SMS notification must be sent to all the guests involved in the event.

- Developed registration for the new user.
- Logging in the user with his username and password.
- Used MongoLab to store the user credentials.
- searching for locations using Yelp API
- Flexibility to login the user with his Google credentials.
- Flexibility to login the user with his Facebook credentials.
- Creating events
- Events are stored into mongolab along with the users username
- A profile page for the user
- Settings page for the user
- Cordova contacts plugin to add the user to mongolab
- Google maps API in order to locate the location of the event.
- User must be able to check his event list by clicking a button and must be
- able to delete the event
- User can update his contacts list by editing and deleting his contacts using
- the username of the contact.
- Messaging between users. users can create the messages using the respective usernames and the messages will be stored in database and the corresponding user can view his messages.
- Creating friends list manually by providing his contact details.
- Our application uses the messaging service(SMS) to send the invitation the user.
- Cordova calendar plugin is used in order to check user's schedule.

.

6.2 Functional Area/Design Unit B

6.2.1 Functional Overview

Software testing is a critical element of software quality assurance and represents the ultimate review of specifications, design and coding. The testing phase involves the testing of system using various test data; Preparation of test data plays a vital role in the system testing. After preparation the test data, the system under study is tested.

Those test data, errors were found and corrected by following testing steps and corrections are recorded for future references. Thus a series testing is performed on the system before it is ready for implementation.

The various types of testing on the system are:

- Unit testing
- Integrated testing
- Validation testing
- Output testing
- User acceptance testing

6.2.2 Impacts

Unit testing

Page	Button	Expected	Output
Home	Search Event	Open browser for	Open browser for
	Create Events	Display add events page	Display add events
	View Events	Display my events page	Display my events
Create Event	Add Friends	User should be able to add a list of friends	User should be able to add a list of
	Add date	User should be able to add date to the event	User should be able to add date to the

	Add time	User should be able to add time to the event	User should be able to add time to the
	Add Event	The events details should be added to the database	The events details should be added to
View Event	Browse for an event	Should display the list of upcoming events	Should display the list of upcoming
	Select a particular	Should display the	Should display the
	Update details	Can update the date and time of the event	Can update the date and time of
	Delete Event	Can delete the	Can delete the
Event Messages	Messages	Should display a window where all the messages related to the event appear.	A window where all the messages related to the event appeared.

6.2.3 Requirements

- Business Rules
- Transaction corrections, adjustments and cancellations
- Administrative functions
- Authentication
- Authorization levels
- Audit Tracking
- External Interfaces
- Certification Requirements
- Reporting Requirements
- Historical Data

7 Open Issues

We have no open issues with my android application. While installing and running eclipse and tomcat we may have some problem because many library's we include. We can solve this in future

8 Acknowledgements

Kanade, Anuradha, Aarthi Gopal, and Shantanu Kanade. "A study of normalization and embedding in MongoDB." Advance Computing Conference (IACC), 2014 IEEE International . IEEE, 2014. I have learned Mongodb and to implement my project form this research paper iam very thankful to the profesors

9 References

[1] http://appshopper.com/social‐networking/isocialite
A native application which helps in socialising with people.
[2] https://developer.foursquare.com/docs/explore#req=users/self
A search API which helps a user to search places.
[3] https://www.yelp.com/developers/documentation/v2/overview
A search API which helps a user to search restaurants or movies or events in a city.
[4] http://codepen.io/ionic/pens/public/

Sample codes provided by ionic in creating pages for native application.

[5] https://play.google.com/store/apps/details?id=com.microsoft.hddl.app&hl=en

[6] Kanade, Anuradha, Aarthi Gopal, and Shantanu Kanade. "A study of normalization and embedding in MongoDB." Advance Computing Conference (IACC), 2014 IEEE International . IEEE, 2014.

[7] https://developers.google.com/maps/documentation/directions/intro

Google directions API to help a user in getting locations..

.

.